TASK FORCE WEB

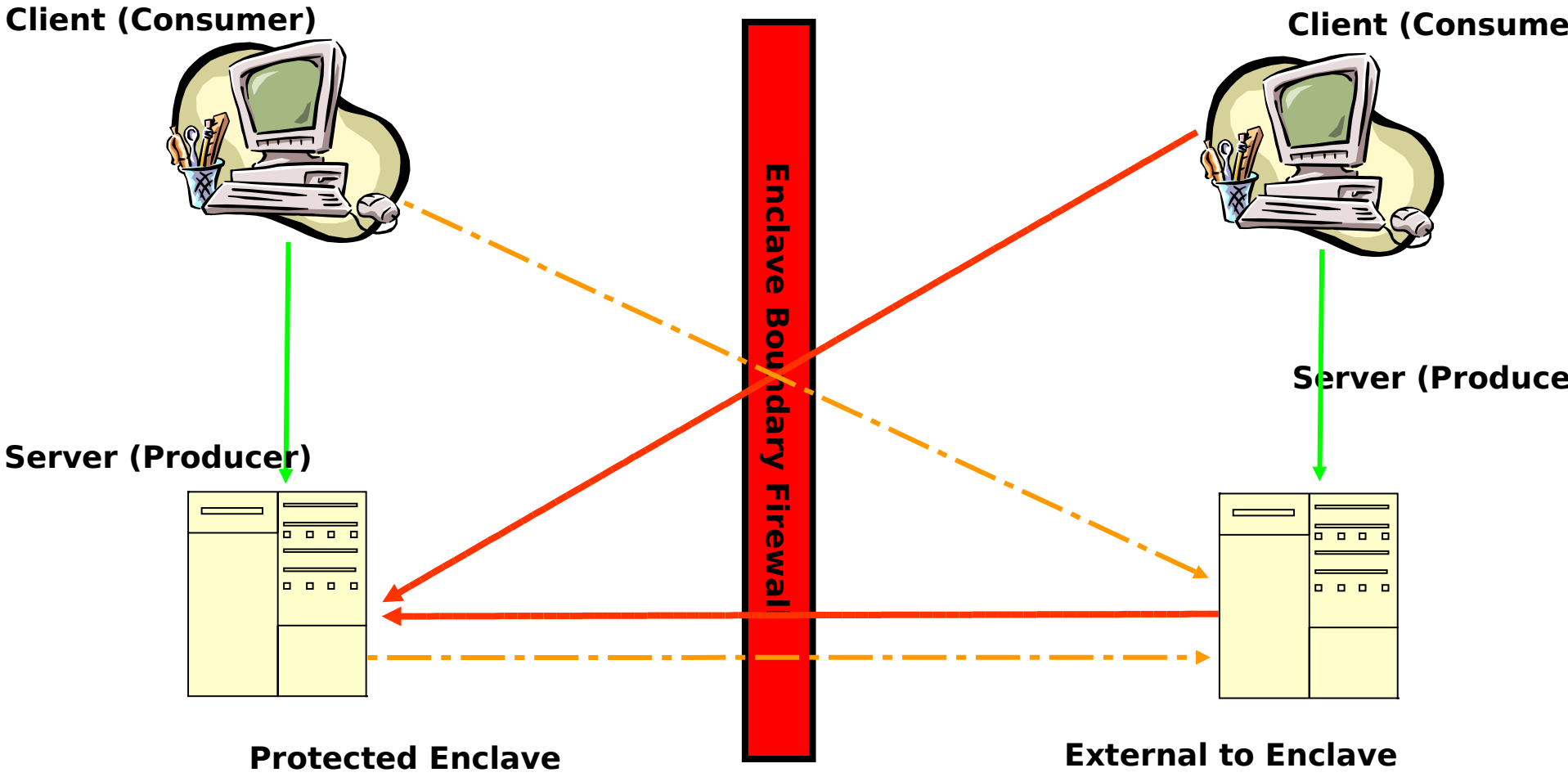
URL Rewrite
9 July 2002

Michael Donovan, EDS
NMCI/ISF Enterprise Technology

Agenda

- URL Rewrite Overview
- The Rewrite Process
- URL Rewrite Limitations
- Optional Strategies

URL Rewrite Overview



URL Rewrite Overview

- Why do we perform URL rewrite?
  - Access to portal transformation services
    - XML/XSL to HTML Rendering (via Xalan)
    - Access Denied/Request for Access Form
    - Portal Request Interface (PRI) header insertion --- which is required for:
      - Portal Look and Feel Integration (via User Selected Template/Style Sheet)
      - Session Management (ID, User, Roles)
      - Application Error Reporting
  - Proxy access to enclave protected applications

URL Rewrite Overview

- *URL Rewrite* is the process by which requests to content integrated into the portal are proxied through the portal.
- Implemented as a portal common service
- For example:
  - A portlet contains the URL https://myapplication.navy.mil/default.asp
  - The URL rewrite process modifies the URL: https://www.homeport.navy.mil/servlet/PortalConnector/user=michael.donovan@https://myapplication.navy.mil/default.asp
- URL Format:
  - <PortalConnector-URL>/<parameters>@<target-URL>

The Rewrite Process

- The processor examines HTML on the portal server, using the Portal Connector. (rewriteURL=Y)
- The processor parses the HTML looking for keywords that may contain URL references.
- When URLs are identified, the processor prepends the PortalConnector URL and parameters to the target URL.
- When a user clicks on a URL, the request is sent through the portlet interface.
- The portal proxies the request to the target URL and processes the response.

The Rewrite Process HTML Keywords

- The processor searches the HTML content of the UFS response for the following (in order):
  - HREF=
  - SRC=
  - URL=
  - BACKGROUND=
  - ACTION=
- These URLs will be rewritten

The Rewrite Process Limitations

- The processor will NOT rewrite URLs under the following conditions:
  1. The URL is embedded between two script tags.
     Example: <script> … src=”http://www.test.com” …</script>
  2. The URL is dynamically generated by client-side script or embedded in objects.
     Example: document.location.href = Variable + “test.jsp”
  3. The URL is a relative link and no BASE_HREF tag is supplied.
     Example: <img src= "/images/mygif2.gif">
  4. The URL is embedded within an event.
     Example: onmouseclick=”… src=’http://www.test.com’ …”
  5. The URL is included as a parameter.
     Example: src=”http://www.test.com?url=www.test2.com
     Note: The URL http://www.test.com will get re-written, but the www.test2.com will not.
  6. If a close tag ‘>’ is located before an open tag ‘<’ prior to the value currently being evaluated.

Examples

Before parsing:

```
<HEAD>
        <TITLE>Page Title</TITLE>
</HEAD>
<body>
        <img src = "http://myapp.navy.mil/images/mygif.gif">

        <img src = "/images/mygif2.gif">

        <a href = "nextpage.html">Next Page</a>
</body>
```

Examples

After Parsing:

```
<HEAD>
   <TITLE>Page Title</TITLE>
</HEAD>
<BODY>
 <img src =
"http://www.homeport.navy.mil/PortalConnector/user=joe@http://myapp.navy.mil/images/mygif.gif">
 <img src =
"http://www.homeport.navy.mil/PortalConnector/user=joe@http://myapp.navy.mil/images/mygif2.gif">
 <a href =
"http://www.homeport.navy.mil/PortalConnector/user=joe@http://myapp.navy.mil/info/nextpage.html"> Next Page</a>
</BODY>
```

Relative URL Parsing

- In certain specific cases of content with relative URLs the Portal will not know the full URL to content. As defined in the HTML standard, an undefined or underivable base will result in an unresolvable URL and a "broken" link.
- As an example, this content may have passed thru the UFS from other Data Oriented Services. In this case the use of the HTML header tag "BASE HREF=" may be required. Refer to the example below.

```
<HEAD>
        <TITLE>Page Title</TITLE>
        <BASE HREF="http://MyDataOrientedService.navy.mil/">
</HEAD>
<body>
        <img src = "/images/mygif2.gif">
</body>
```

Relative URL Parsing

- Relative URLs are handled in accordance with the following standards:
  - RFC 2616 Hypertext Transfer Protocol – HTTP/1.1
  - RFC 1808 Relative Uniform Resource Locators
  - W3C HTML 4.01 - 24 December 1999 (section 12)

Optional Strategies

- URL Rewrite is NOT necessary if:
  - URL can be resolved from anywhere in the enterprise
  - Users are entirely within an enclave and do not need access from outside the enclave
- And:
  - Access to other portal services is not required (XML Parsing, PRIData, etc.)
- Also, an optional forward proxy mechanism is under investigation.

TASK FORCE WEB

XML Transformation

09 July, 2002

Michael Donovan, EDS
NMCI/ISF Enterprise Technology

Agenda

- XML Transformation Overview
- XML Transformation Process

XML Transformation Overview

- Experience has shown that various implementations of XSLT rendering implementations produce different results.
- The XML Transformation process implements a standard rendering solution to support standardized rendering.

XML Transformation Process

- The UFS returns XML data with an imbedded XSLT reference to the portal.
- The XML is identified by the response mime-type or XML formatting cues.
- The data is passed to the rendering engine.
- The result is returned for further processing.
- The XML Rendering Engine can be called as a common service or is used as part of normal processing.